

REMARKS

In the Office Action mailed December 8, 2006, the Examiner provisionally rejected claims 1-27, 29, and 30 on the ground of nonstatutory obviousness-type double patenting, as being unpatentable over claims 1-25 of copending Application No. 09/760,031; rejected claims 13-24 under 35 U.S.C. § 101 as being directed to non-statutory subject matter; and rejected claims 1-27, 29, and 30 under 35 U.S.C. 103(a) as being unpatentable over USPN 5,999,948 to Nelson et al. ("Nelson"), in view of USPN 6,510,352 B1 to Badavas et. al. ("Badavas"). In response to the Final Office Action, Applicant respectfully requests reconsideration in view of the foregoing claim amendments and remarks below. Claims 1-4, 6-16, 18-26 and 29-30 are pending in the application of which claims 1, 13, 25, and 26 are independent. By this amendment, Applicant cancels claims 5, 17, 27 and 28 without prejudice or disclaimer.

Claims 1-4, 6-8, 10-16, 18-20, 22-26, and 29-30 are amended in the foregoing amendments to further clarify Applicant's invention and to place the claims in condition for allowance. Claims 31-38 are new. Support for the amendments and new claims may be found in Applicant's specification and accordingly, no new matter is introduced by the claim amendments or new claims. Specifically, for example, support for the amendments to claim 1 can be found generally in FIG. 2A-2E in Applicant's specification, where FIG. 2A illustrates exemplary class describing data structures and associated option describing data structures, and figures 2B-2E illustrate exemplary listing data structures. Support may also be found on page 13 at line 24-26, stating "when these definitions are compiled, the compiler generates an OptionBinding object

corresponding to each local-option declaration in the program” and on page 14 at line 15, stating “The compiler also generates an object... to describe each class.”

Response to Arguments

The Examiner contends that “the Specification fails to explicitly disclose that ‘compiling’ is the process by which a compiler translates program code into machine language, or object code. See Office Action at page 4. Applicant disagrees. The Specification states, “in typical data processing systems, a compiler reads source code and creates the object code which is ultimately processed.” See Background of the Invention. Moreover, Applicant asserts that it is not necessary to disclose the definition of a term that is known in the art, such as “compiling.” See MPEP § 2164.01 (“A patent need not teach, and preferably omits, what is well known in the art. *In re Buchner*, 929 F.2d 660, 661, 18 USPQ2d 1331, 1332 (Fed. Cir. 1991); *Hybritech, Inc. v. Monoclonal Antibodies, Inc.*, 802 F.2d 1367, 1384, 231 USPQ 81, 94 (Fed. Cir. 1986), cert. denied, 480 U.S. 947 (1987); and *Lindemann Maschinenfabrik GMBH v. American Hoist & Derrick Co.*, 730 F.2d 1452, 1463, 221 USPQ 481, 489 (Fed. Cir. 1984).”)

The Examiner contends that the specification at page 11, shows the program compiling by getting and setting an appropriate value and that Nelson performs a similar operation. However, at page 11, Applicant explains, “the compiler **generates code** for each of the fundamental option constructs by translating as follows....” See Specification at page 11, lines 5-7. The compiling of Applicant’s claims refers to the generation of code as well as other compiler operations described in the claims. Although Nelson, at col. 12, discloses reading and writing values when a button is pushed in the application, the reading and writing disclosed by Nelson in col. 12 is a

process performed at **run time**. Nowhere does Nelson disclose or suggest **compilation** as recited in Applicant's claims.

The Examiner notes that Badavas discloses Process Control Objects (PCOs) that can have mandatory parts and optional parts and that the optional parts can be added subsequent to creation. See Office Action at page 4. This disclosure by Badavas also does not disclose the compiling of Applicant's claims. Nowhere does Badavas disclose or suggest how definitions of PCOs are compiled, or how operations on the optional parts are compiled.

Provisional Nonstatutory Obviousness-type Double Patenting

In the Office Action mailed March 20, 2006, the Examiner provisionally rejected claims 1-27, 29, and 30 under the judicially created doctrine of obviousness-type double patenting over claims 1-25 of copending Application No. 09/760,031, and maintains this rejection in the Office Action dated December 8, 2006. Applicant requests that the Examiner holds the double patenting rejection in abeyance until the claims in the instant application are otherwise allowable.

Rejection under 35 U.S.C. § 101

Although Applicant disagrees with the Examiner regarding the conclusion that claim 13 is non-statutory, Applicant has amended claim 13 in the interest of expediting examination and allowance of the instant claims. Claim 13 has been amended according to the Examiner's suggestion and now includes memory and a processor. See Office Action at page 3.

Rejection under 35 U.S.C. § 103

The Examiner rejected claims 1-27, 29, and 30 under 35 U.S.C § 103(a) as being unpatentable over Nelson in view of Badavas. See Office Action at page 6. The independent claims 1, 13, 25, and 26 were rejected under the same rationale. See Office Action at page 6-9. As noted above in the Response to Arguments section of this paper, Applicant disagrees that Nelson at Col. 12, lines 61-67 discloses the “compiling an operation on an option value.”

Nelson does not disclose or suggest, “compiling an operation on a value associated with a selected option in an instance of the class, wherein compiling the operation comprises searching the at least one option describing data structure associated with the class describing data structure to locate the type description corresponding to the selected option, and using the type description to type check the operation,” as recited in Applicant’s claim 1. Nelson also does not disclose or suggest, “compiling a definition of a class which supports options... wherein compiling the definition comprises generating a class describing data structure, wherein the class describing data structure is associated with at least one option describing data structure, the at least one option describing data structure comprising a type description corresponding to each option defined in the class or through the class inheritance hierarchy,” as recited in Applicant’s claim 1. As noted above, Nelson’s reading and writing at Col. 12 is not equivalent to the aforementioned element of Applicant’s claim 1.

Badavas also does not disclose or suggest **how** a definition of a class which supports options would be compiled, or how an operation on an option value in an instance of the class would be compiled and therefore does not make up for the

deficiency of Nelson. Since neither Nelson nor Badavas disclose or suggest the "compiling" elements of Applicant's claim 1, the rejection under 35 U.S.C. § 103(a) should be withdrawn. See MPEP § 2143.03 ("To establish prima facie obviousness of a claimed invention, all the claim limitations must be taught or suggested by the prior art. *In re Royka*, 490 F.2d 981, 180 USPQ 580 (CCPA 1974).").

For the foregoing reasons, the rejection claims 13, 25, and 26, which recite similar elements directed to "compiling" should be withdrawn. Dependent claims 2-4, 6-16, 18-26, and 29-38, which depend from claims 1, 13, 25, and 26 are allowable for at least the same reasons.

In view of the foregoing amendments and remarks, Applicant respectfully requests reconsideration and reexamination of this application and the timely allowance of the pending claims.

Please grant any extensions of time required to enter this response and charge any additional required fees to our deposit account 06-0916.

Respectfully submitted,

FINNEGAN, HENDERSON, FARABOW,
GARRETT & DUNNER, L.L.P.

Dated: MAY 18, 2007

By: 

Maura K. Moran
Reg. No. 31,859